CLAIMS

What is claimed is:

1. A method for dynamically binding a user interface to information, comprising:

specifying with a first language a first action;

specifying with a second language a first data source associated with the first action;

rendering output with a third language based at least partially on the first action;

wherein the second language is embedded in the first language; and

wherein the first action can set or get the first data source.

2. The method of claim 1 wherein:

the first language allows for the specification of JavaServer Page action elements.

3. The method of claim 1 wherein:

the second language is based on the Javascript language.

4. The method of claim 1 wherein:

the first data source identifies one of: 1) an object field; 2) an object property; and 3) an Extensible Markup Language document element.

5. The method of claim 4 wherein:

an object is a JavaBean.

6. The method of claim 1 wherein:

the first data source is one of: 1) an array; 2) a list; 3) a map.

7. The method of claim 1 wherein:

the third language can include at least one of: Hypertext Markup Language (HTML), Dynamic HTML, Extensible HTML, and Extensible Markup Language.

8. The method of claim 1 wherein:

the first action can be a child of another action.

9. The method of claim 1 wherein:
the first action can have at least one child action.

10. The method of claim 9 wherein:
the at least one child action can have at least one other child action.

11. The method of claim 9 wherein:
the at least one child action can selectively process the first data source.

12. The method of claim 9 wherein:
the at least one child action can refer to the first data source with a context defined by the first action.

13. The method of claim 9 wherein:
the at least one child action can perform at least one of the following actions on the first data source: 1) set; 2) get; 3) sort; and 4) filter.

14. The method of claim 9, further comprising:
rendering a list or a table based on the first data source.

15. A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:
specify with a first language a first action;
specify with a second language a first data source associated with the first action;
render output with a third language based at least partially on the first action;
wherein the second language is embedded in the first language; and
wherein the first action can set or get the first data source.

16. The machine readable medium of claim 15 wherein:
the first language allows for the specification of JavaServer Page action elements.

17. The machine readable medium of claim 15 wherein:
the second language is based on the Javascript language.

18. The machine readable medium of claim 15 wherein:
the first data source identifies one of: 1) an object field; 2) an object property; and 3) an Extensible Markup Language document element.

19. The machine readable medium of claim 18 wherein:
an object is a JavaBean.

20. The machine readable medium of claim 15 wherein:
the first data source is one of: 1) an array; 2) a list; 3) a map.

21. The machine readable medium of claim 15 wherein:
the third language can include at least one of: Hypertext Markup Language (HTML), Dynamic HTML, Extensible HTML, and Extensible Markup Language.

22. The machine readable medium of claim 15 wherein:
the first action can be a child of another action.

23. The machine readable medium of claim 15 wherein:
the first action can have at least one child action.

24. The machine readable medium of claim 23 wherein:
the at least one child action can have at least one other child action.

25. The machine readable medium of claim 23 wherein:
the at least one child action can selectively process the first data source.

26. The machine readable medium of claim 23 wherein:

the at least one child action can refer to the first data source with a context defined by the first action.

27. The machine readable medium of claim 23 wherein:

the at least one child action can perform at least one of the following actions on the first data source: 1) set; 2) get; 3) sort; and 4) filter.

28. The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

render a list or a table based on the first data source.

29. A computer data signal embodied in a transmission medium, comprising:

a code segment including instructions to specify with a first language a first action;

a code segment including instructions to specify with a second language a first data source associated with the first action;

a code segment including instructions to render output with a third language based at least partially on the first action;

wherein the second language is embedded in the first language; and

wherein the first action can set or get the first data source.

30. A software framework for rendering at least one object on a user interface, comprising:

a first language capable of specifying a first action;

a second language capable of specifying a first data source associated with the first action;

a third language capable of rendering output based at least partially on the first action;

wherein the second language is embedded in the first language; and

wherein the first action can set or get the first data source.

31. The framework of claim 30 wherein:

the first language allows for the specification of JavaServer Page action elements.

32. The framework of claim 30 wherein:
the second language is based on the Javascript language.

33. The framework of claim 30 wherein:
the first data source identifies one of: 4) an object field; 2) an object property; and 3) an Extensible Markup Language document element.

34. The framework of claim 33 wherein:
an object is a JavaBean.

35. The framework of claim 30 wherein:
the first data source is one of: 4) an array; 2) a list; 3) a map.

36. The framework of claim 30 wherein:
the third language can include at least one of: Hypertext Markup Language (HTML), Dynamic HTML, Extensible HTML, and Extensible Markup Language.

37. The framework of claim 30 wherein:
the first action can be a child of another action.

38. The framework of claim 30 wherein:
the first action can have at least one child action.

39. The framework of claim 38 wherein:
the at least one child action can have at least one other child action.

40. The framework of claim 38 wherein:
the at least one child action can selectively process the first data source.

41. The framework of claim 38 wherein:

the at least one child action can refer to the first data source with a context defined by the first action.

42. The framework of claim 38 wherein:

the at least one child action can perform at least one of the following actions on the first data source: 4) set; 2) get; 3) sort; and 4) filter.

43. The framework of claim 38 wherein:

a list or a table can be rendered based on the first data source.